Lab 5
October 23, 2017
Spectral Analysis

INSTRUCTIONS:
All lab submissions include a written report and source code in the form of an m-file. The report contains all plots, images, and figures specified within the lab. All figures should be labeled appropriately. Answers to questions given in the lab document should be answered in the written report. ***The written report must be in PDF format***. Submissions are done electronically through Compass 2G.

1 Discrete Fourier Transform (DFT)

A discrete (sampled) signal can be analyzed on a computer using the DFT. It can be seen that DTFT enables representation of $x(n)$ in terms of complex exponentials, namely $e^{i\omega n}$. $X_d(\omega)$ is continuous. However, on a computer we require that $X_d(\omega)$ be discrete. Since ω is periodic with respect to 2π, if we let $\omega = 2\pi k/N$ for $k = 0, \ldots, N-1$ where N is the length of $x(n)$, then we obtain the discrete fourier transform (DFT), which be written as

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-i2\pi kn/N}, \text{ for } k = 0, \ldots, N-1 \tag{1}$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{i2\pi kn/N}, \text{ for } n = 0, \ldots, N-1 \tag{2}$$

The DFT is essentially a sampled version of the DTFT.

The DFT can be expressed as a matrix-vector product $X = Zx$ where x and X are input and output vectors, respectively, and Z is the DFT matrix whose elements are defined by $e^{-i2\pi kn/N}$ where k is the row index and n is the column index. The computational complexity of DFT is $\mathcal{O}(N^2)$.

Report Item 1: Implement a function called **myMatrixDFT** that implements the DFT using matrix-vector multiplication by constructing the DFT matrix. Validate your function with **fft**. Based on your implementation, why does it make sense that the DFT is $\mathcal{O}(N^2)$?

2 Using the FFT

From here on, we will be using MATLAB's built-in **fft**. The FFT is a $\mathcal{O}(N \log N)$ implementation of the DFT. In this section, we will learn how to use **fft** to analyze

signals. Assume that you have a signal $x(t) = \cos(2\pi f_0 t + \phi)$ with frequency f_0 and a phase ϕ. The sampled version of $x(t)$ is $x(n) = \cos(2\pi(f/f_s)n + \phi)$. Let $f_0 = 100$Hz, $N = 512$, and $\phi = \pi/4$ such that $x(t)$ looks like Figure 1.

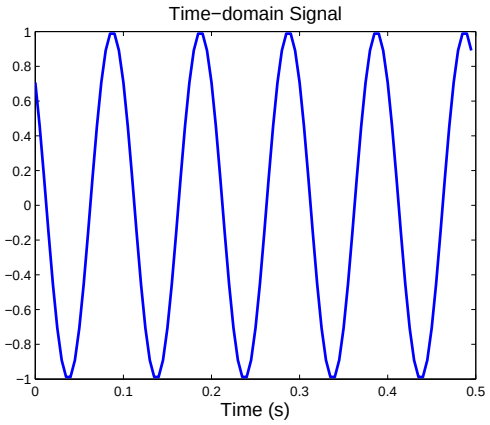
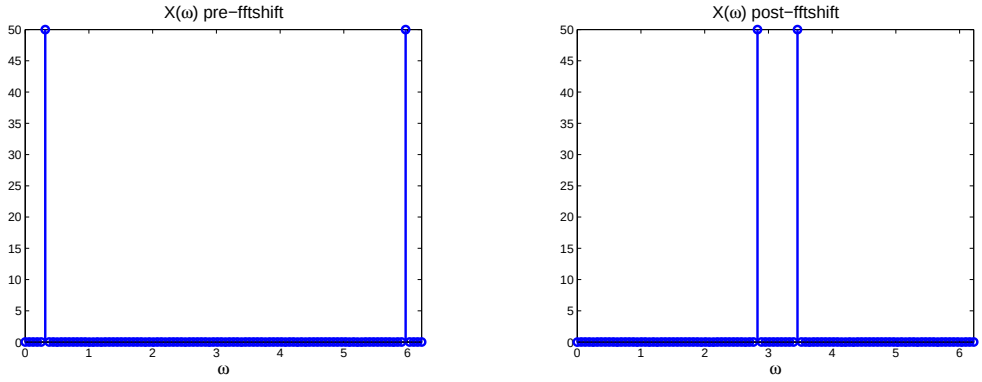


Figure 1

Taking the **fft** of $x(n)$ gives us Figure (2a). The coefficients returned by **fft** are ordered by frequencies correspond to $\omega = 2\pi k/N$ for $k = 0, \ldots, N-1$. The valid range of ω is $-\pi$ to π. From the periodicity of $X_d(\omega)$, we know that $X_d(\omega)$ is identical between $(-\pi, 0)$ and $(\pi, 2\pi)$. Hence, by swapping the first and second halves of $X_d(\omega)$, we can get a vector that corresponds to $X(\omega)$ within the valid range of $(-\pi, \pi)$. However, the size of each half depends on N being even or odd and must be done carefully. This can be accomplished using **fftshift** on the output of **fft**.



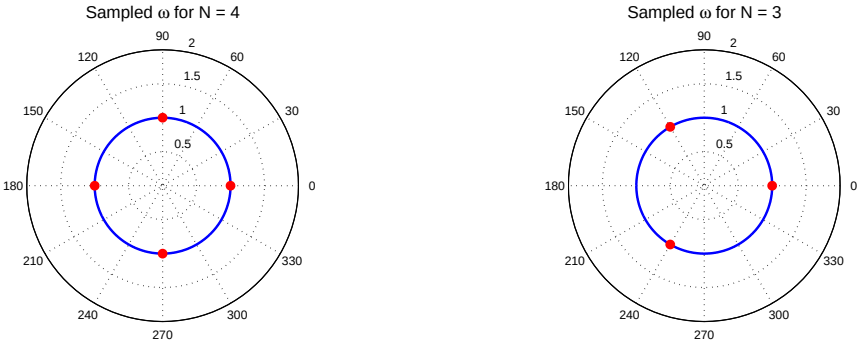
(a) (b)

After applying **fftshift** to the DFT coefficients, a similar operation must be done for ω. The valid range for ω can be obtained using the following piece of code:

```
w = fftshift((0:N−1)/N*2*pi);
w(1:N/2) = w(1:N/2) − 2*pi;
```

Figure 3: Code that generates ω within the "valid" range for a signal of length N.

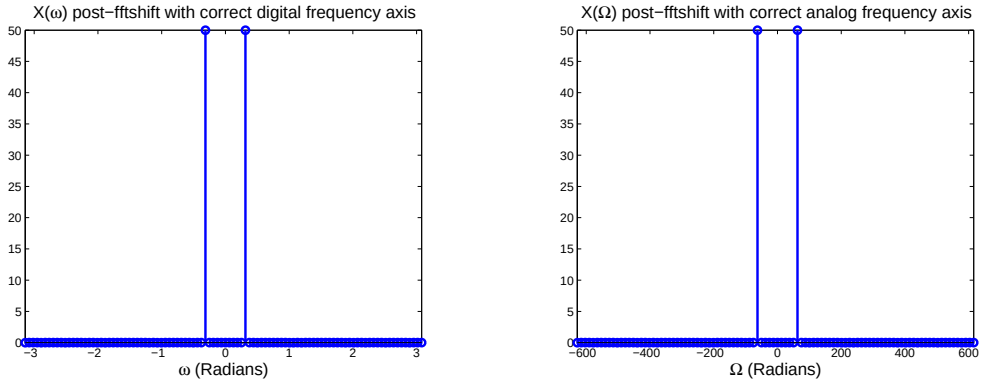
The reason we use this piece of code instead of **linspace**$(-\pi, \pi, N)$ is due to the way the DFT is sampled. When N is odd, the endpoints $-\pi$ and π are not actually included. When N is even, only $-\pi$ is included while π is not. Arbitrary use of **linspace** is a common pitfall for DSP students!



(a) (b)

Figure 4: Sampling of ω corresponding to $X_d(\omega)$ between $-\pi$ and π. When N is odd, the endpoints $-\pi$ and π are not included. When N is even, only $-\pi$ is included.

The result after performing **fftshift** on both the output of the **fft** and the ω axis is shown in Figure 5.



(a) (b)

Figure 5

Report Item 2: Load *signal.mat*. A variable called x should appear in your workspace. Using **fft**, find $X(\omega)$ and plot its magnitude and phase. Given that $f_s = 200$ Hz, find $X_d(\Omega)$ and plot the magnitude and phase. How many frequency tones are there? What are their values (in Hz)?

If **fftshift** is performed on the output of **fft**, one must first perform an **ifftshift** on the frequency coefficients prior to taking the **ifft**.

3 Zero-Padding

Zero-padding is the addition of zeros to the end of a discrete signal. This increases the number of frequency samples in the DFT such that it bears a closer resemblance to the DTFT. The improvement in visual quality can enable one to discern more peaks in the frequency spectrum. Zero-padding does ***not*** add additional information to the frequency spectrum. Given a signal $x(n)$ of length $N < M$, zero-padding the signal to length M is equivalent to inserting $M - N$ zeros at the end of the signal prior to taking its DFT. This changes the sampling of ω from $2\pi k/N$ over $k = 0, \ldots, N-1$ to $2\pi k/M$ over $k = 0, \ldots, M-1$. The additional zeros do not alter the original Fourier coefficients.

Report Item 3: Load *NMRSpec.mat*. The sampling frequency of the NMR signal is $f_s = 2000$ Hz. Plot the magnitude and phase spectrum of the signal (*hint:* MATLAB's **fft** can perform zero-padding). Calculate the 32-point DFT spectrum of the first 32 points of the signal. Can you distinguish the peaks corresponding to creatine (around 209 Hz) and chlorine (185 Hz)? Zero-pad the signal to 512 points and plot the resulting magnitude and phase spectrum. Does zero-padding help? Why or why not?

4 Truncation and Windowing

To analyze a signal on a computer, the length of the signal must be finite and discrete. This requires sampling and truncation. Sampling in time domain leads to periodicity in the frequency domain. Truncation is multiplication of the time domain signal with a windowing function. Consider the following windowing functions:

- Rectangular Window

$$w(n) = \begin{cases} 1, & n = 0, \ldots, N-1 \\ 0, & \text{else} \end{cases} \tag{3}$$

- Triangular Window

$$w(n) = \begin{cases} \frac{2n}{N}, & n = 0, \ldots, N/2 \\ w(N-n), & n = N/2+1, \ldots, N-1 \\ 0, & \text{else} \end{cases} \tag{4}$$

- Hamming window

$$w(n) = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{N}\right), & n = 0, \ldots, N-1 \\ 0, & \text{else} \end{cases} \tag{5}$$

- Hanning window

$$w(n) = \begin{cases} 0.5 - 0.5\cos\left(\frac{2\pi n}{N}\right), & n = 0, \ldots, N-1 \\ 0, & \text{else} \end{cases} \tag{6}$$

- Kaiser window

$$w(n) = \begin{cases} I_0\left(\beta\sqrt{1-((n-\frac{N}{2})/\frac{N}{2})^2}\right), & n = 0, \ldots, N-1 \\ 0, & \text{else} \end{cases} \tag{7}$$

  $I_0(x)$ is the zeroth-order modified Bessel function of the first kind. The choice of β affects mainlobe and sidelobe heights.

Report Item 4: For $N = 20$, plot the rectangular window using **stem** and its magnitude response (in decibels) zero-padded to 512 points using **plot**. Use **mag2db** to convert the magnitude response into decibels. Repeat for triangular, hamming, hanning, and Kaiser windows. For the Kaiser window, let $\beta = 0.1$. How do the mainlobe width and sidelobe height of the rectangular window compare to the triangular window? Between the rectangular window and hamming window, which one has lower side lobes?

The length of $w(n)$ affects the width of the mainlobe. This can be seen from the analytical expression of a rectangular window. The DTFT of the rectangular window of length N is

$$X(\omega) = \sum_{n=0}^{N-1} e^{-i\omega n} = \frac{1-e^{-i\omega N}}{1-e^{-i\omega}} = \frac{e^{-i\omega N/2}}{e^{-i\omega/2}} \frac{\sin(N\omega/2)}{\sin(\omega/2)} \tag{8}$$

Equation (8) can be expressed using the **diric** function (named after Dirichlet) which is the periodic version of the **sinc** function. MATLAB defines the **diric** function as

$$D(x) = \begin{cases} \frac{\sin(Nx/2)}{N\sin(x/2)}, & x \neq 2\pi k \text{ for integer } k \\ (-1)^{k(N-1)}, & x = 2\pi k \text{ for integer } k \end{cases} \tag{9}$$

Report Item 5: Using **diric**, implement the DTFT of a rectangular window of length $N = 20$. Plot the magnitude and phase response (do not convert to decibels). Repeat for $N = 40$. For each case, what is the width of the mainlobe (in radians)? Measure from null-to-null.

Windowing is performed by multiplying $x(n)$ with $w(n)$ where $x(n)$ is the original signal and $w(n)$ is a windowing function. The effects of windowing can be understood from several points of view. By knowing the shape of the frequency response of $w(n)$ and using the fact that multiplication in the time domain is equivalent to convolution in the frequency domain, one can visualize the effects that each window has on the frequency response of a signal. An alternative point of view can be gleaned in the time domain.

From (2), it is implied that the IDFT is periodic with respect to N. The implications of this periodicity are illustrated below. The sine wave in Figure 6a has the expected magnitude response in Figure (6b). Namely, two peaks that resemble impulses. However, if we take a few more additional samples, some additional peaks appear in the spectrum. This is known as *spectral leakage*.

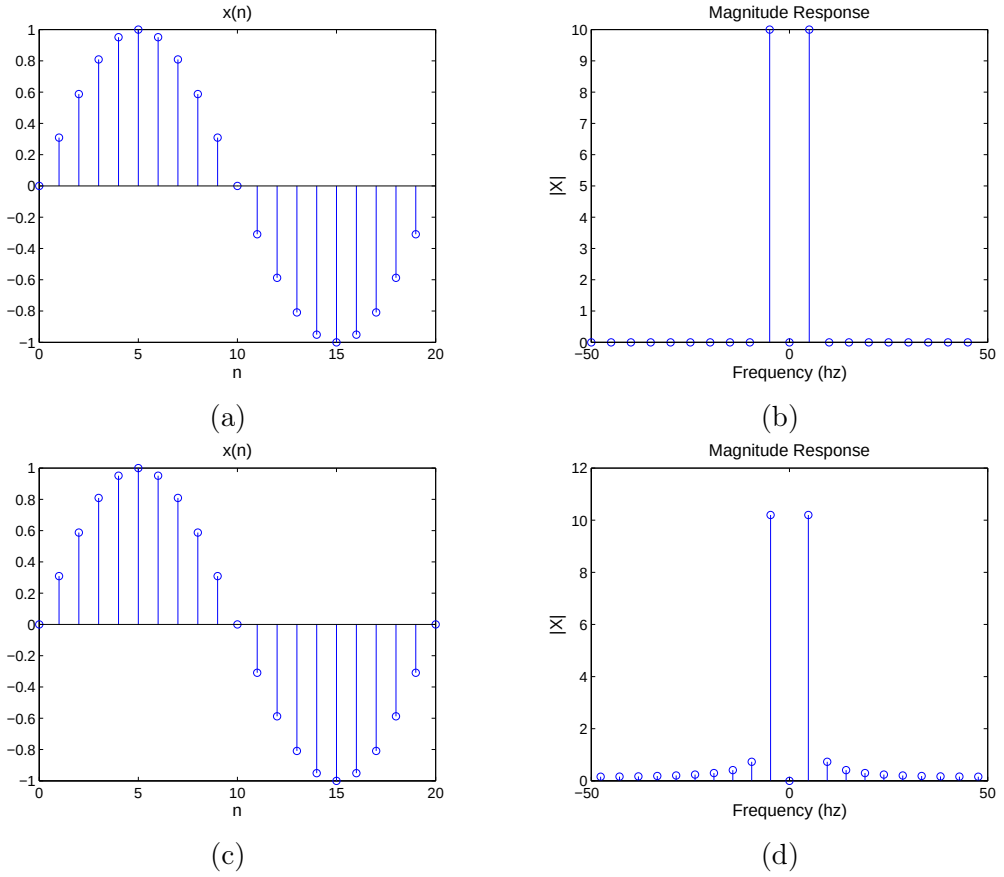
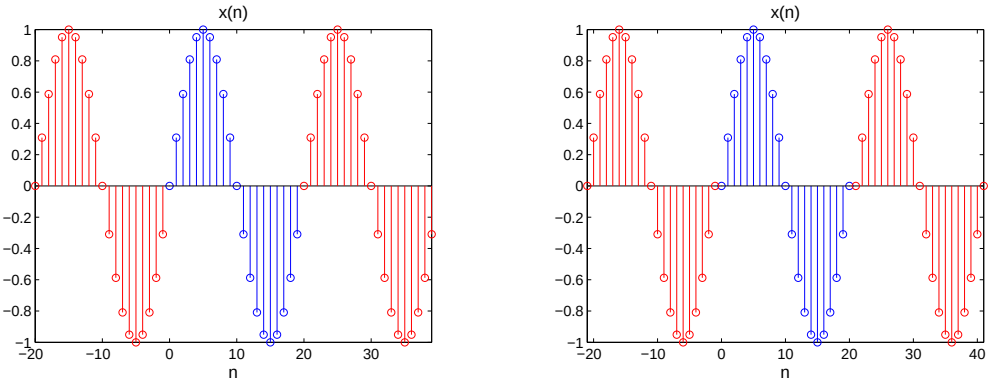


Figure 6

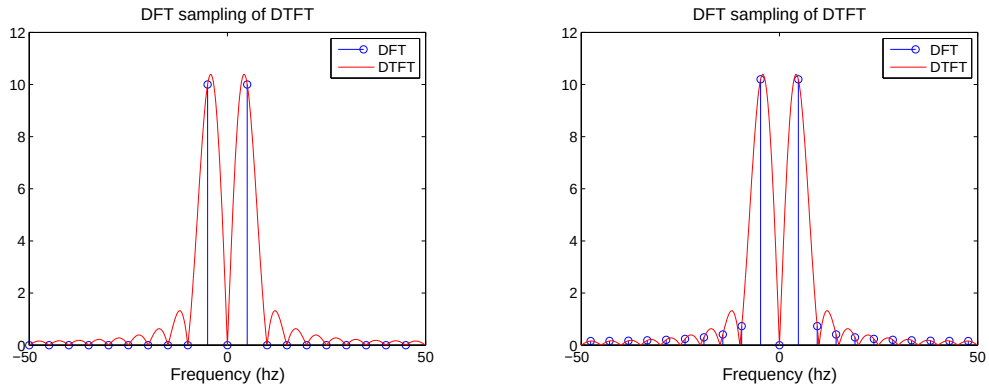
Even though Figures 6a and 6c look identical, to the DFT they are not. The DFT sees $x(n)$ as a single period in the periodic signal. From the DFT's point of view, Figures 6a and (6c) resemble Figures 8a and 9, respectively. Recall that in the definition of DFT, the signal is assumed to be periodic with respect to N, where N is the number of samples. In 6c, there are 21 samples instead of 20 in 6a. However, we are expecting the signal to be periodic with 20 samples. The result is what looks like a small discontinuity as seeen in Figure 8a (20 samples) and 9 (21 samples).



(a) (b)

Figure 7: The input to the DFT is considered to be a single period of a periodic signal. The DFT point of view of 6a and 6c is shown in 8a and 9, respectively.

It can be said that this discontinuity is responsible for the high frequency sidelobes that appear in 6d. But in a way, the sidelobes are always there. The DFT is the sampled version of the DTFT and the DTFT of a truncated sinusoid is a pair of Dirichlet functions. The chosen N coincides with sampling of the frequency domain that lands right on the mainlobes and none of the sidelobes of the Dirichlet functions. The fact that there's all this activity going on in the frequency domain and our sampling happens to miss all of it is known as the *picket fence* effect. This effect can be mitigated by zero-padding.



(a) (b)

Figure 8

Report Item 6: Given $x(n) = \sin(2\pi f_0 n \Delta t)$ where $f_0 = 5$, $\Delta t = 0.02$, and $n = 0, \ldots, N-1$. What value(s) must N be in order to minimize spectral leakage? Plot the magnitude and phase response of $x(n)$ using the minimum value of N.

5 Spectrogram

In this problem, you will learn how to make a spectrogram with the spectrogram function in MATLAB (we will do this instead of making you compute the spectro-

gram by hand). A spectrogram displays a temporal-frequency representation of a signal, represented as a matrix. The algorithm is essentially as follows:

(1) Given x[n], chunk x[n] into N (i=1,2, . . . N) blocks of length L

(2) Apply a windowing function to the block from (1)

(3) Take a block from step (2) and compute the DFT of the block

(4) Store the output of (3) into the i^{th} column of an output matrix

The above steps are abstracted away from you in MATLAB.
The syntax for spectrogram in MATLAB is:
[s,w,t] = spectrogram(x,window,noverlap,w)

An example spectrogram is seen below in figure 9.

Report Item 7: Here are the following steps:

1.) Load song1.mat. The variable song1 should appear in your workspace

2.) Listen to song1 by the sound command. Set Fs to 44100 (second argument of sound())

3.) Use the spectrogram function with following arguments
a. x is the song signal (yes, just song1)
b. window is a Hamming window of size 4096
c. noverlap is 2048
d. w is 4096
4.) The output is stored in s,w,t. Type in the following command: imagesc(t,w,log(abs(s)));

The output is a matrix, with the y-axis representing digital frequencies between 0 and pi. The x-axis representing each block. The y-axis and x-axis are the frequency and temporal representations of the signal

5.) Repeat 1-4, for song2 and song3

6.) Please turn in a spectrogram for each song. For each spectrogram, comment on the frequency distribution as a function of blocks. Can you make out the drums, chords, synthesizers from looking at the spectrogram. Does it make sense? NOTE : The three genres are French Electronic, Trap, and Math Rock.

0.25

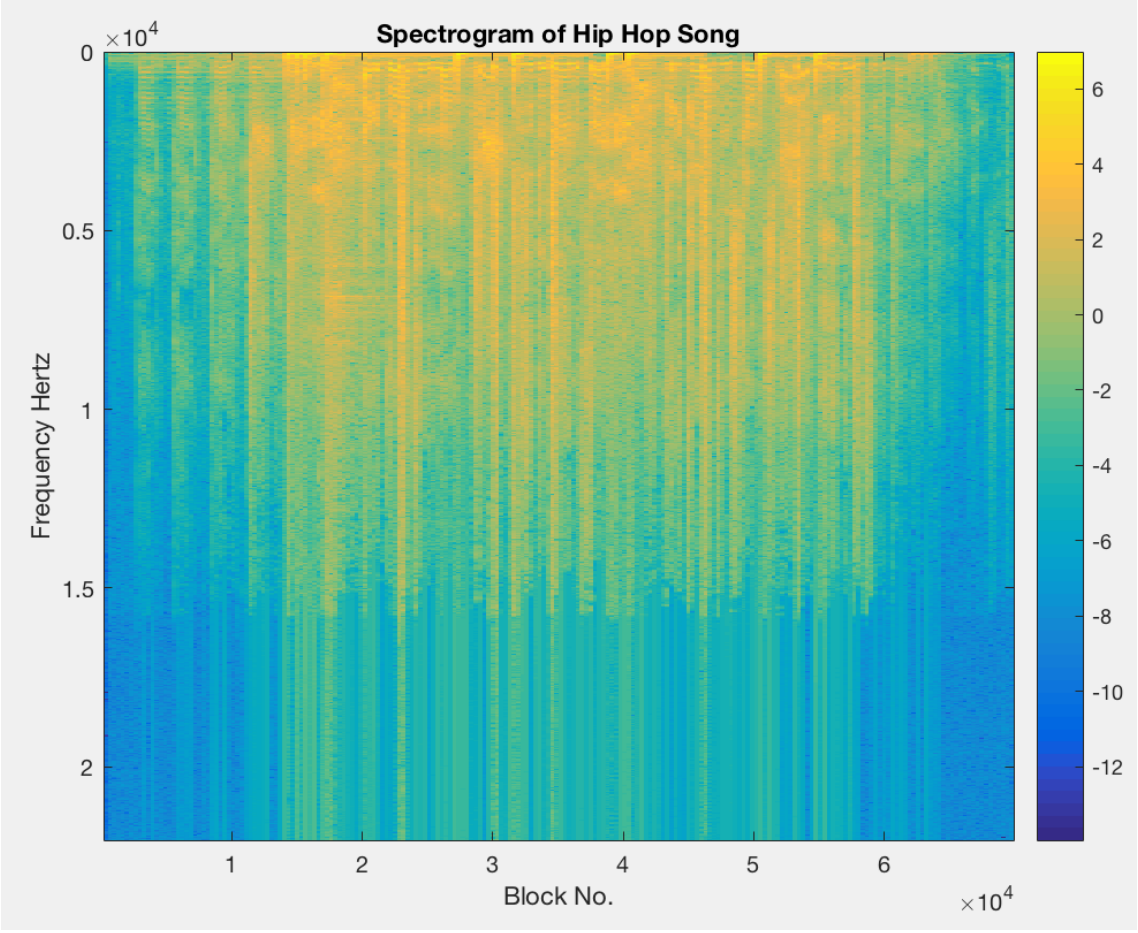


Figure 9: Spectrogram Example